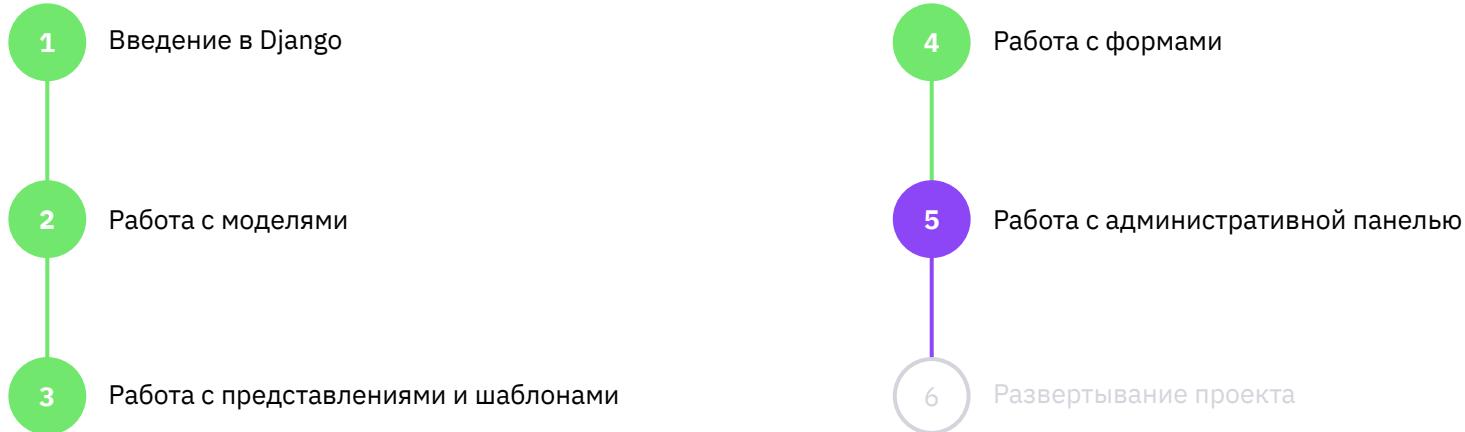


# Работа с административной панелью

Урок 5



# План курса





# Содержание урока





## Что будет на уроке сегодня

- 📌 Узнаем об административной панели Django
- 📌 Разберёмся в настройке админ панели
- 📌 Изучим добавление пользовательских моделей в панель
- 📌 Узнаем о способах персонализации админ панели



Что такое  
административная  
панель в Django?





## Административная панель

**Административная панель** в Django — это готовый интерфейс, который позволяет управлять данными вашего приложения. Она предоставляет возможность создавать, редактировать и удалять записи в базе данных, а также осуществлять множество других действий, связанных с управлением приложением.





# Настройка административной панели





## Первоначальные настройки “админки”

Адрес по которому мы можем зайти в админ панель был настроен при создании проекта. Проверьте urls.py проекта

```
from django.contrib import admin
...
urlpatterns = [
    path('admin/', admin.site.urls),
    ...
]

```

```
LANGUAGE_CODE = 'ru-ru'
```



## Создание суперпользователя

По умолчанию доступ к административной панели имеет только суперпользователь (superuser). А если забыли пароль, его можно изменить, но не подсмотреть

```
python manage.py createsuperuser  
python manage.py changepassword <username>
```



## Стандартная админка

Раздел “Пользователи и группы” доступен по умолчанию. Администратор сайта может делать всю цепочку CRUD: создавать, просматривать, изменять и удалять с группами и пользователями.



### Пользователи

- Добавить пользователя
- Просмотр и редактирование пользователя
- Удаление пользователя



### Группы

- Добавить группу
- Просмотр и редактирование группы
- Удаление группы



Добавление  
пользовательских моделей  
в административную панель





## Создание моделей

Любые модели, которые мы хотим подключить к административной панели создаются в `models.py`. Далее создаются и применяются миграции:

```
python manage.py makemigrations  
python manage.py migrate
```

```
from django.db import models  
  
class Category(models.Model):  
    name = models.CharField(max_length=50, unique=True)  
    ...
```



## Подключение моделей к административной панели

Открываем файл admin.py приложения. Его создал Django в процессе выполнения команды startapp. Файлы admin.py всех приложений объединяются в одной “админке”

```
from django.contrib import admin
from .models import Category
...
admin.site.register(Category)
...
```



Персонализация моделей  
в административной панели





## Изменение модели



### Добавляем новые поля

прописываем параметр default для существующих записей



### Создаём миграции

- 1) задаём значения по умолчанию полям без default для существующих записей
- 2) Возвращаемся к изменению полей модели



### Применяем миграции

```
python manage.py migrate
```



## Настройка списка изменения

Рассмотрим основные возможности админки в рамках списка значений.  
Что и как можно изменить под себя?

- 💡 Отображение дополнительных полей – `list_display`
- 💡 Сортировка строк – `ordering`
- 💡 Добавление фильтрации – `list_filter`
- 💡 Текстовый поиск – `search_fields`, `search_help_text`
- 💡 Добавление новых действий – `@admin.action`, `actions`



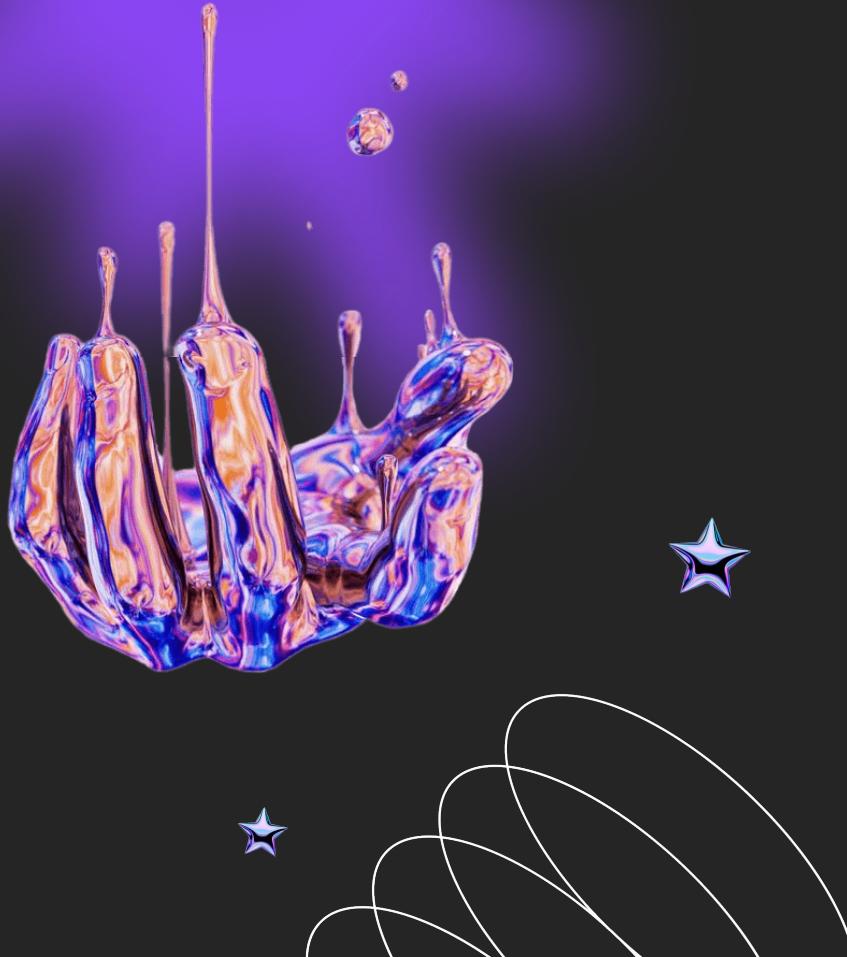
## Настройка изменения отдельной записи

Рассмотрим основные возможности админки в рамках изменения отдельной записи. Что и как можно изменить под себя?

- 💡 Отображение полей – `fields`
- 💡 Отображение полей только для чтения – `readonly_fields`
- 💡 Детальная настройка отображения полей – `fieldsets`



# Итоги занятия





## На этой лекции мы

- 📌 Узнали об административной панели Django
- 📌 Разобрались в настройке админ панели
- 📌 Изучили добавление пользовательских моделей в панель
- 📌 Узнали о способах персонализации админ панели



## Задание



1. Для закрепления материалов лекции попробуйте самостоятельно набрать и запустить демонстрируемые примеры.
2. \*Загляните в официальную документацию Django и изучите дополнительные возможности работы с административной панелью.



Спасибо за внимание

